CMPT 310 Artificial Intelligence Survey

Assignment 2 Report

Maoshun Shang

301280479

Instructor: James Delgrande

Map Coloring Problem algorithm:

The map coloring problem is solved by implementing a basic backtrack search algorithm. It involves with selecting country and try to match a color to the selected country with checking the consistency. Additionally, along with the basic backtrack search, the algorithm integrated with variable selecting method: minimum remaining value (MRV) and degree heuristic; color selecting method: least constraining value. The testing of the entire algorithm is done on a graph with 10 vertices. It prints out the solution with a runtime of the solution.

High level description of the algorithm:

The function "solve" is a recursive function, it calls itself to solve the next node if the current layer successfully found a color for the current vertex selected without breaking the constraints. For each iteration of the "solve" function, it selects a variable by either in the order of 'G', minimum remaining value (MRV), or degree heuristic.

1. Minimum remaining value: the function calls a subroutine to select a variable that have least number of remaining value available in its domain.
2. Degree heuristic: the function calls a subroutine to traverse the graph 'G' to select the highest degree node. By the adjacency list, the node with maximum length is the one that having most number of neighbors.

Then, in the stage of finding a matching color, the function either try testing consistency by the order of color or try in the order that put the least constraining value to the neighbor vertices. In either case, the function calls itself for coloring the rest of the vertices if a consistent color is found, or return '-1' to inform the parent function that the lower layer function results in no color to match with constraints are kept.

1. By order of color: the function uses a for loop to test consistency from color 1 to color k, for the first color that does not violate the consistency, the tuple (vertex number, color) is added to the assignment list.
2. By least constraining value: the function calls a subroutine to calculate for each of the color, how many choices it rules out for the neighboring vertices. Then sort by the number of choices by ascending order and try from the least constraining one.

Testing and Runtime Comparison:

There are two test cases: one with no solution and one has a solution:

The 1st test case (no solution):

The first testing is done on a graph with 10 vertices and 3 colors, because this test case has no solution (The program will print “FAIL”). Therefore, the heuristics will show the benefits of using them in terms of the reduction on running time. The runtime results are shown in milliseconds:

	Order in ‘G’ (no heuristics)	MRV	Degree
Color by order (no heuristics)	3.617	1.839	0.629
Color by least constraining value	17.547	2.131	1.866

Note: the runtime significantly depends on the design of algorithms and the performance of machine.

The function also prints out the number of inconsistency met under the no-solution test case. The number of inconsistency solved until the algorithm determines there is no solution are: 951 for none of the heuristic functions used while 135 for heuristic functions are used.

<u>The 2nd test case:</u>

The second test case uses the same graph but with four colors. By 4-coloring theorem, there is always a solution under this scenario.

The backtracking algorithm solved all the consistency with similar runtime and number of inconsistency solved. The benefits of using heuristic functions under a “has-solution” case will be shown on a larger graph G. In order to keep simplicity, the test case involving higher number of vertices are not included in the testing stage.

In general, the three types of heuristic functions have their own advantages:

1. Degree heuristic is the most efficient function because of the lower time complexity. It also decreases the branching factor because the vertices are chosen in the order of degree.
2. The minimum remaining value (MRV) choose the vertex that most likely to fail to speed up the search. The heuristic prunes search at an early stage.
3. The least constraining value heuristic gives the neighboring vertices more color options in their domain. Therefore, it leaves the maximum flexibility for the adjacent vertices.

Conclusion:

The effort of heuristic functions is shown in the case that no solution exists in the input graph. By comparing horizontally, the degree heuristic is the most efficient function to reduce the running time and inconsistencies. Overall, the three heuristic functions reduced the number of times that the program runs into an inconsistent state. Therefore, it saves the backtracking search time.